https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

Design of Tuning Mechanism of PID Controller for Application in three Phase Induction Motor Speed Control

Alfred A. Idoko[1], Iliya. T. Thuku[2], S. Y. Musa[3], Chinda Amos[4]

[1, 2, 3] Department of Electrical and Electronics Engineering, Modibbo Adama University of Technology Yola, Adamawa State, Nigeria.

[4] Nigeria. National Centre for Technology Management, Modibbo Adama University of Technology Yola, Adamawa State, Nigeria.

Abstract**—** *This paper presents a design of tuning mechanism of Proportional Integral Derivative Controller for application in three phase induction motor speed controls. It demonstrates, in detail, how to employ the MatLab tool so as to search efficiently for the optimal PID controller parameters within a mechanism system. The proposed approach has superior features, including: easy implementation; stable convergence characteristics; and less computational effort. Three phase induction motors has complex mathematical modelling which makes it difficult to design the speed controller. Software PID Tuning Mechanismwas developed herein and used to obtain both the initial PID parameters under normal operating conditions and the optimal parameters of PID control under fully-loaded conditions. The proposed PID controller Tuning Mechanism will automatically tune its parameters within these ranges. In order to prove the performance of the proposed tuning mechanism for the PID controller, a three phase asynchronous motor was modelled in MATLAB, the transfer function was obtained using the software and a controller was designed using PID. The modelling and simulations results show the potential of the proposed controller to be very efficient.*

Keywords— PID Controller, Modelling of Induction Motor, Design of PID Tuning Software.

I. INTRODUCTION

Speed control of an induction motor has been a major challenge since its discovery by a French physicist François Arago in 1824. The induction motor is made up of two distinct parts, stator which is a stationary part and rotor (also called armature core windings) which is the rotary part. Induction motor is an alternating current electric machine which by excitation, produces a torque[1]. The torque result from an electromagnetic induction from the magnetic flux due to the stator core windings as a result of rotation about a fixed position in the stator core windings [2].

Induction motor is the most common electrical machine used in modern industries. It has gained such popularity due to its various advantages. The various advantages are high efficiency, low cost, good self-starting, simplicity in design, the absence of the collector brooms system, and a small inertia. Even though it has a wide range of advantages it also has quite few disadvantages. Induction motor has disadvantages, such as complex, multivariable and nonlinear mathematical model and is not inherently capable of providing variable speed operation [3].

During the past decades, process control techniques in industries have made great advances. Numerous control methods such as: adaptive control; neural control; and Fuzzy control have been studied [4]. Among these the best known is the Proportional Integral Derivative (PID) controller, which is being widely used because of its simple structure and robust performance within a wide range of operating conditions. Unfortunately, it has been quite difficult to properly tune the gains of PID controllers because many industrial plants are often burdened with problems such as: high orders; time delays; and nonlinearities [5].

Over the years, several heuristic methods have been proposed for the tuning of PID controllers. The first method used the classical tuning rules proposed by Ziegler and Nichols [6]. In general, it is often hard to determine optimal or near optimal PID parameters with the Ziegler–Nichols formula for many industrial plants [7]. For this reason it is highly desirable to increase the capabilities of PID controllers by adding new features. Many Artificial Intelligence (AI) techniques have been employed to improve the controller performances for a wide range of plants whilst retaining their basic characteristics. AI techniques such as: neural networks; fuzzy systems; and

https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

neural-fuzzy logic have been widely applied to the proper tuning of PID controller parameters [8].

Software tool, is one of the modern tuning mechanisms. The software tool can generate a high-quality solution for shorter calculation time and stable convergence characteristics than other stochastic methods [4]. Much research is still in progress for proving the potential of the software tool for solving complex induction motor system operation problems. Because the software tool method is an excellent tuning mechanism and a promising approach to solving the PID controller parameters tuning problem. This controller is called the software-tuning PID controller. However, the PID controller is not robust to wide parameter variation and large external disturbance [11]. This serves especially for highly coupling nonlinear system where the PID controller lacks adaptive capability. The difficulty in obtaining transfer function of three phase induction motor poses a serious challenge to this mechanism.

However, if the PID controller parameters are chosen incorrectly, the controlled process input can be unstable, i.e., its output diverges, with or without oscillation, and is limited only by saturation or mechanical breakage. Instability is caused by excess gain, particularly in the presence of significant lag [1].

Generally, stabilization of response is required and the process must not oscillate for any combination of process conditions and setpoints, though sometimes marginal stability (bounded oscillation) is acceptable or desired [1].

Mathematically, the origins of instability can be seen in the Laplace domain. The total loop or close loop transfer function is:

$$H_{(S)} = \boldsymbol{K_d} + \frac{\boldsymbol{K(s)G(s)}}{\boldsymbol{1+K(s)G(s)}} \quad \ldots (1)$$

Where.

K(s): PID transfer function

G(s): Plant transfer function

The system is called unstable where the closed loop transfer function diverges for some **s**.This happens for situations where K(s) G(s) = -1. Typically, this happens when there is a 180 degree phase shift. Stability is guaranteed when K(s)G(s) < 1 for frequencies that suffer high phase shifts.

In order to achieve practical requirement, engineers have to adjust the parameters under different operating conditions and device a means of obtaining complex transfer function. However, the robustness is limited to a small range. A mechanism to overcome this disadvantage is called the software-tuning mechanism. The parameter tuning at any instance is usually based on a structurally fixed mathematical model produced by a software procedure [11]. Unfortunately, recent plants find it difficult to obtain their fixed mathematical models. This paper proposes a software-tuning mechanism for PID controller for an application in three phase induction motors. At the same time, the robustness will be expanded to a large range. In this paper, a practical high-order mechanism system with a PID controller is adopted to test the performance of the proposed software-tuning mechanism for PID controller. As compared with the other mechanisms such as the one proposed by [9][8] [7][6], it is found that the nominal values and tuning ranges of the PID parameters by the proposed software-tuning mechanism can be accurately determined. MatLab simulations and experimental results have shown that the proposed controller is most suitable and robust for PID controller.

II. MATERIALS AND METHODS

In this work the following materials were used: Software tools, PID Controller block in MATLAB package, Induction Motor and Voltage source.

2.1 PID Controller

PID controllers are widely used in industries for speed control purpose. A PID controller calculates an "error" value as the difference between the measured process value and the desired set point. The PID controller calculation involves three separate constants and is accordingly sometimes called three-term control i.e. the proportional, integral and the derivative value which is denoted by PID as show in figure 1 below.

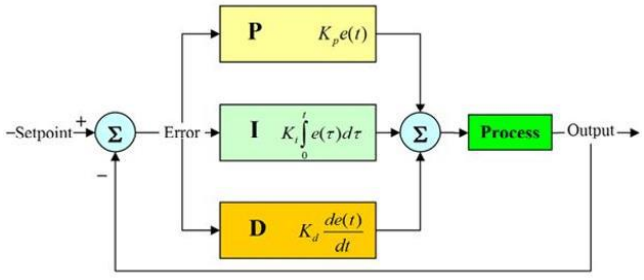

Fig. 1: Block diagram of PID controller [10]

Tuning a control loop is the adjustment of its control parameters (proportional band/gain, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response [11]. Stability (no unbounded oscillation) is a basic requirement, but beyond that, different systems have different behavior, different applications have different requirements, and requirements may conflict with one another [4].

https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

PID tuning is a difficult problem, even though there are only three parameters and in principle is simple to describe, because it must satisfy complex criteria within the limitations of PID control. There are accordingly various methods for loop tuning, and more sophisticated techniques are the subject of patents; this section describes some traditional manual methods for loop tuning. Figure 2 below shows the effect of tuning the PID parameters on the process response.

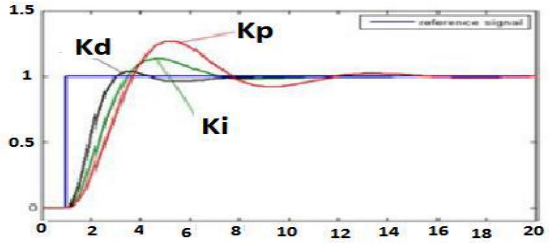

Fig. 2: Process variables for different Kp, Ki and Kd values [11].

Tuning the Proportional Term:Process variable for different Kp values (Ki and Kd held constant) is shown in Figure 2. The proportional term makes a change to the output that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain [11]. The proportional term is given by the Equation (12)

$P_{out} = K_p e^{(t)}$ .... (2)

$e = G / Kp$ … (3)

Integral Term: Process variable for different Ki values (Kp and Kd held constant) is shown in figure 2. The contribution of the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously [12]. The accumulated error is then multiplied by the integral gain (Ki) and added to the controller output. The integral term is given by the Equation (4)

$I_{out} = ki\int e^{(\tau)}\, d_\tau$ … (4)

Derivative Term:Process variable for different Kd values (Ki and Kp held constant) is shown in Figure 2. The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain Kd. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain Kd [12]. The derivative term is given by the Equation (5)

$D_{out} = K_d + \frac{d}{dt}c(t)$ … (5)

2.2 PID Tuning Mechanisms

There are several methods for tuning a PID Controller. Tuning methods with its advantages and disadvantages are given in Table 1.

Table.1: PID Tuning Method [11]

Method	Advantage	Disadvantage
Manual Tuning	No maths required. On-line method	Require experience personnel
Ziegler-Nichols	Proven method. Online method	Process upset, some trial-and-error
Software tools	Consistent tuning. Online or offline method. Can support Non-Steady State Tuning	some cost and training involved
Cohen-Coon	Good process models	some math. Offline method only good for first order processes.

The most effective methods generally involve in the development of some form of the process model, and then choosing P, I, and D based on the dynamic model parameters. Manual tuning methods can be relatively inefficient, particularly if the loops have response times on the order of minutes or longer. The choice of method will depend largely on whether or not the loop can be taken "offline" for tuning, and the response time of the system. If the system can be taken offline, the best tuning method often involves subjecting the system to a step change in input, measuring the output as a function of time, and using this response to determine the control parameters [11].

2.3 Software Tool

Most modern industrial facilities no longer tune loops using the manual calculation methods shown above. Instead, PID tuning and loop optimization software are used to ensure consistent results. These software packages will gather the data, develop process models, and suggest optimal tuning. Some software packages can even develop tuning by gathering data from reference changes [13].

https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

Advances in automated PID Loop Tuning software also deliver algorithms for tuning PID Loops in a dynamic or Non-Steady State (NSS) scenario. The software will model the dynamics of a process, through a disturbance, and calculate PID control parameters in response.

2.4 Modelling of Induction Motor

In the control of any power electronics drive system, mathematical model of the plant is required [14]. From the objective equation (1), the Plant transfer function [8] is required to design the loop controller. To obtain the Induction Motor (Plant) transfer function and design any type of controller to control its speed, mathematical model is required. The Equivalent circuit of induction motor in d-q axis fig. (3) will be used to analyze the mathematical modeling of induction motor.

The induction motor can be analyzed as a three phase stator winding magnetically coupled with the rotors ones, independently on the rotor typology.

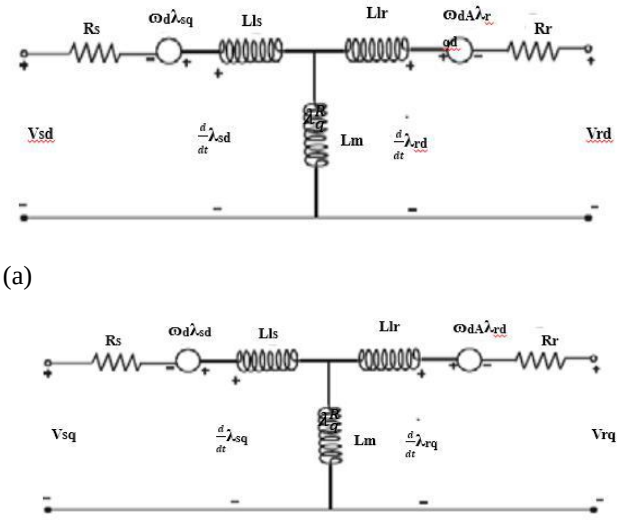

Fig. 3: Equivalent circuit of induction motor (a) d-axis and (b) q-axis[15]

The equivalent circuit used for obtaining the mathematical model of the induction motor is shown in the Fig. (3) The induction motor model is established using a rotating (*d*, *q*) field reference (without saturation) concept. An induction motor model is then used to predict the voltage required to drive the flux and torque to the demanded values within a fixed time period. This calculated voltage is then synthesized using the dynamic model of the motor in Simulink.

The induction motor can be analyzed as a three phase stator winding magnetically coupled with the rotors ones, independently on the rotor typology.

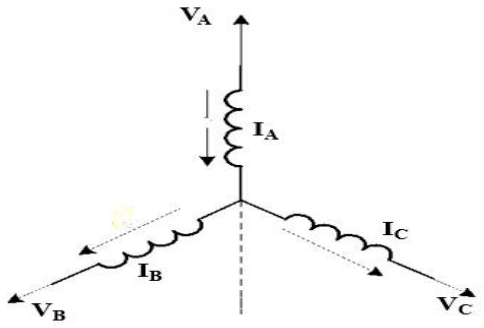

Fig. 4: Phasor diagram of Stator winding [10]

$V_A = p\lambda_A + R_S I_A$ … (6)

$V_B = p\lambda_B + R_S I_B$ …. (7)

$V_C = p\lambda_C + R_S I_C$ … (8)

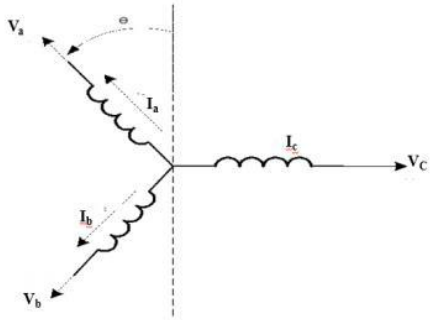

Fig. 5: Phasor diagram of Rotor winding [10]

$Va = p\lambda a + RSIa$ … (9)

$Vb = p\lambda b + RSIb$ … (10)

$Vc = p\lambda c + RSIc$ … (11)

When the dynamic model of the induction motors is determined, we can transform the three-phase machine into a two-phase machine using an orthonormal transformation matrix T:

$$T = \begin{bmatrix} \sqrt{\frac{2}{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \quad \text{… (12)}$$

$$\begin{bmatrix} \lambda ABC \\ \lambda abc \end{bmatrix} = \begin{bmatrix} Mss & Msr \\ Mrs & Mrr \end{bmatrix} \begin{bmatrix} IABC \\ Iabc \end{bmatrix} \quad \text{… (13)}$$

$$M_{SS} = \begin{bmatrix} Lo + Lsds & -\frac{Lo}{2} & -\frac{Lo}{2} \\ -\frac{Lo}{2} & Lo + Lsds & -\frac{Lo}{2} \\ -\frac{Lo}{2} & -\frac{Lo}{2} & Lo + Lsds \end{bmatrix} \quad \text{… (14)}$$

https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

$$M_{rr} = \begin{bmatrix} Lo + Lrdr & -\frac{Lo}{2} & -\frac{Lo}{2} \\ -\frac{Lo}{2} & Lo + Lrdr & -\frac{Lo}{2} \\ -\frac{Lo}{2} & -\frac{Lo}{2} & Lo + Lrdr \end{bmatrix} \quad \dots (15)$$

$$Msr = \begin{bmatrix} Lo\cos\theta & Lo\cos(\theta + 120°) & Lo\cos(\theta + 240°) \\ Lo\cos(\theta + 240°) & Lo\cos\theta & Lo\cos(\theta + 120°) \\ Lo\cos(\theta + 120°) & Lo\cos(\theta + 240°) & Lo\cos\theta \end{bmatrix} \dots (16)$$

Where Mss, Mrr and Msr are the stator mutual inductance, rotor mutual inductance respectively. Lo is the no load inductance.

Below equations represent the equation of the equivalent two phase machine. For control purpose, another transformation is required.

$V_{\alpha}^{s} = R_s i_{\alpha}^{s} + p\lambda_{\alpha}^{s}$ … (17)

$V_{B}^{s} = R_s i_{B}^{s} + p\lambda_{B}^{s}$ … (18)

$V_{0}^{s} = R_s i_{0}^{s} + p\lambda_{0}^{s}$ … (19)

$V_{\alpha}^{R} = R_s i_{\alpha}^{R} + p\lambda_{\alpha}^{R}$ …(20)

$V_{B}^{R} = R_s i_{B}^{R} + p\lambda_{B}^{R}$ … (21)

$V_{0}^{R} = R_s i_{0}^{R} + p\lambda_{0}^{R}$ … (22)

If we want to move these equations in a reference frame fixed with the stator, we have to apply a rotation matrix around the Z axis [16].

$\lambda_{\alpha}^{s} = (L_{ds} + \frac{3}{2}L_0) i_{\alpha}^{s} + \frac{3}{2}L_0 \cos(\theta)\, i_{\alpha}^{R} - \frac{3}{2}L_0 \sin(\theta)\, i_{B}^{R} \dots$ (23)

$\lambda_{B}^{s} = (L_{ds} + \frac{3}{2}L_0) i_{\alpha}^{s} + \frac{3}{2}L_0 \cos(\theta)\, i_{\alpha}^{s} - \frac{3}{2}L_0 \sin(\theta)\, i_{B}^{s} \dots$ (24)

$\lambda_{\alpha}^{R} = (L_{ds} + \frac{3}{2}L_0) i_{\alpha}^{R} + \frac{3}{2}L_0 \cos(\theta)\, i_{\alpha}^{s} - \frac{3}{2}L_0 \sin(\theta)\, i_{B}^{s} \dots$ (25)

$\lambda_{B}^{R} = (L_{ds} + \frac{3}{2}L_0) i_{\alpha}^{R} + \frac{3}{2}L_0 \cos(\theta)\, i_{\alpha}^{s} - \frac{3}{2}L_0 \sin(\theta)\, i_{B}^{s} \dots$ (26)

With this transformation, the final equation in a reference system called ‘’stator dq0 reference frame’ are:

$\lambda_{d}^{s} = L_s i_{d}^{s} + L_m i_{d}^{R}$ ... (27)

$\lambda_{q}^{s} = L_s i_{q}^{s} + L_m i_{q}^{R}$ ... (28)

$\lambda_{d}^{R} = L_R i_{d}^{R} + L_m i_{d}^{s}$ … (29)

$\lambda_{q}^{R} = L_R i_{q}^{R} + L_m i_{q}^{s}$ … (30)

$V_{d}^{s} = R_s i_{d}^{s} + p\lambda_{d}^{s}$ … (31)

$V_{q}^{s} = R_s i_{q}^{s} + p\lambda_{q}^{s}$ ... (32)

$0 = R_r i_{d}^{R} + p\lambda_{d}^{R} + w\lambda_{q}^{R}$ … (33)

$0 = R_r i_{q}^{R} + p\lambda_{q}^{R} + w\lambda_{d}^{R}$ … (34)

Torque (T) $= \lambda_{q}^{R} i_{d}^{R} + \lambda_{q}^{R} i_{d}^{R}$ … (35)

Where Ls and Lr are the respective stator and rotor inductance, Rs and Rr are the respective stator and rotor resistance. I_d and I_q are the respective d and q axis of the stator and rotor winding.

Using these equations, it is possible to build the block scheme in order to simulate the machine. The blocks are: Stator Voltage Transformation, Fluxes, Current, Electromagnetic, and Mechanical Subsystem.

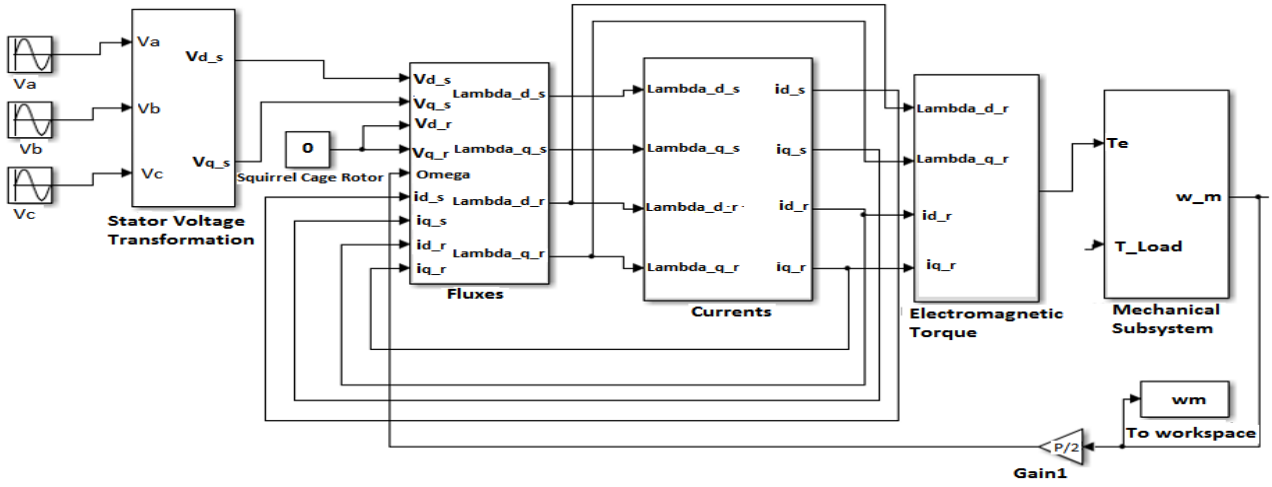

Fig. 6: Three Phase Induction Machine Asynchronous Motor Dynamic Model

2.5 Modelling PID Controller

Fig.(1) above shows the block diagram of PID Controller. The first term A proportional (TOP), an integral (CENTER) and A derivative (BOTTOM). A proportional controller may not give steady state error performance which is needed in the system. An integral controller may give steady state error performance but it slows a system down. So the addition of a derivative term helps to cure both of these problems. The

https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

proportional, integral and derivative outputs are added together. The PID controller can be thought of as having a transfer function. The PID transfer function can be obtained by adding the three terms together.

$$PID(s) = K_P + \frac{K_i}{s} + Sk_d \quad \cdots (36)$$

The transfer function can be combined into a pole-zero form. Hence, the equation becomes;

$$PID(s) = [sK_p + K_i + s^2K_d]/s \quad \cdots (37)$$

From fig.(1) the output of PID controller *u (t)*, is equal to the sum of three signals: The signal obtained by multiplying the error signal by a constant proportional gain *KP*, plus the signal obtained by differentiating and multiplying the error signal by constant derivative gain *KD* and the signal obtained by integrating and multiplying the error signal by constant internal gain *KI*,. The output of PID controller is given by equation (38), taking Laplace transform, and solving for transfer function, gives ideal PID transfer function given by equation (39) [10]

$$\mathbf{u(t)} = \boldsymbol{K_P e^{(t)} + K_P \frac{de(t)}{dt} + K_I \int e(t)dt \Leftrightarrow U(s) = k_p E(s)s + k_I E(s)\frac{1}{s}} \quad \ldots (38)$$

$$\boldsymbol{U(s) = E(s)\left[K_p + \frac{K_I}{S} + K_D s\right]} \quad \cdots (39)$$

$$\boldsymbol{G_{PID(S)} = K_P + \frac{K_I}{S} + K_D s = \frac{K_D s^2 + K_P s + K_I}{S} = \frac{K_D\left[s^2 + \frac{K_P}{K_D}s + \frac{K_I}{K_D}\right]}{S}.} \quad \cdots (40)$$

Equation (40) is second order system, with two zeros and one pole at origin, and can be expressed to have the following form:

$$\boldsymbol{G_{PID} = \frac{K_D(S + Z_{PI})(S + Z_{PD})}{S} = K_D(S + Z_{PI})\frac{(S + Z_{PD})}{S} = G_{PD}(S)G_{PI}(S)} \quad \ldots (41)$$

Which indicates that PID transfer function is the product of transfer functions PI and PD, Implementing these two controllers jointly and independently will take care of both controller design requirements? The transfer function given by Equation (40 and 41), can also be expressed to have the form:

$$\boldsymbol{G_{PID} = \frac{K_D(S + Z_{PI})(S + Z_{PD})}{S} = \frac{K_D s^2 + (Z_{PI} + Z_{PD})K_D s + (Z_{PI}Z_{PD}K_D)}{S}} \quad \ldots (42)$$

Rearranging the above equation (42) we have:

$$\boldsymbol{G_{PID} = \frac{K_D s^2}{S} + \frac{(Z_{PI} + Z_{PD})K_D s}{S} + \frac{(Z_{PI}Z_{PD}K_D)}{S} = (Z_{PI} + Z_{PD})K_D + \frac{(Z_{PI}Z_{PD}K_D)}{S} + K_D s} \quad \cdots (43)$$

Let, $\boldsymbol{K_1 = (Z_{PI} + Z_{PD})K_D, K_2 = (Z_{PI}Z_{PD}K_D), K_3 = K_D}$.

Then, equation (43) becomes,

$$\boldsymbol{G_{PID} = K_1 + \frac{K_2}{S} + K_3 s} \quad \cdots (44)$$

Since PID transfer function is a second order system, it can be expressed in terms of damping ratio and undamped natural frequency to have the following form:

$$\boldsymbol{G_{PID} = K_P\left[1 + \frac{1}{T_1 S} + T_D s\right] = K_P \frac{T_1 T_D s^2 + T_1 s + 1}{T_1 s}} \cdots (45)$$

Where the integral time $T_I = \frac{K_P}{K_I}$, the derivative time $\boldsymbol{T_D = \frac{K_D}{K_P}, K_I = \frac{K_P}{T_I}, K_D = K_P T_D}$

2.6 Design of PID Tuning Software

The tuning mechanism used is the software tuning mechanism. The mechanism is designed using MatLab tools capable of deriving the transfer function of a complex Induction machine and varying the PID parameters to control the speed of the machine. Below is the various steps to designing the tuning mechanism.

Step 1. Modelling of the three phase induction motor in MatLab.

The three phase induction motor in MatLab is shown in fig.(6) and the model is converted to a block named Motor as shown in fig.(7) below

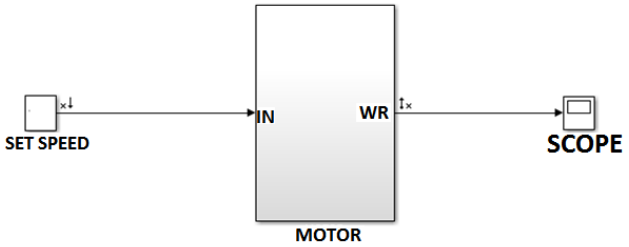

Fig. 7: Induction motor model block

Step 2. Obtain the transfer function of the Motor using MatLab script. The motor is run in MatLab and the step response is analyzed. The analysis further yields the step response of the motor as shown in figure 8 below.

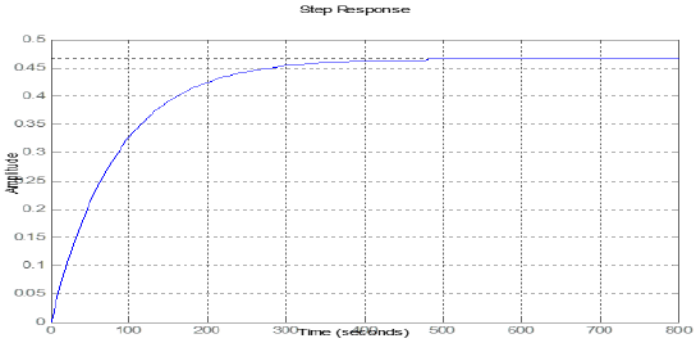

Fig. 8: step response of induction motor

And the transfer function of the step response in figure 8 above is obtained by taking the linear analysis in MatLab.

https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

From input "Torque" to output "Subsystem":

$$Gs = \frac{-4.208e\text{-}12\ s + 0.2598}{S^2 + 45.99\ s + 0.2981} \quad \ldots (46)$$

Linearization at model initial condition Continuous-time transfer function.

Equation 46 represent the transfer function of the three phase Asynchronous induction motor under analysis. This method can be used to obtain the transfer function of any system.

Step 3. Using this transfer function to design a PID controller in MatLab script and testing it on the step response in figure 8 above. Varying the PID parameter (Kp, Ki, Kd) on the script and running the script tunes the step response. This controller is then tuned for optimum response and the parameters are recorded for the actual controller design in the next step.

Step 4. Connect a PID controller to the Motor. Sequel to step 3 above, a PID controller block is connected with the motor block as show in fig.(9) below.

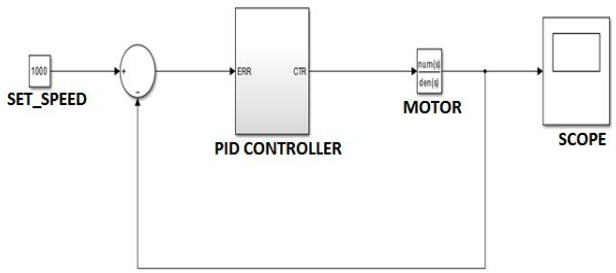

Fig. 9: Three Phase Induction Motor with PID Controller

As shown in figure 9, the output speed of the motor is feedback to the input and the summer computes the error value. This error is being fed to the PID controller and by tuning the PID using software script, the error is compensated. The value of the PID parameters and the motor parameters are programed into the system automatically. However, the script "IM_Initialazation" can be run over the network from a remote location to the plant. With the availability of internet access, this tuning mechanism offers the most advanced software tool for PID Controller in an application to three phase induction motors.

2.7The Induction Motor Parameters

The induction motor used in this simulation is a 50 Hp, 420V, 60Hz, asynchronous motor having the parameters listed in table 2 below. The induction motor stator is fed by a current controlled three-phase source. The stator currents are regulated by hysteresis regulator which generates inverter drive signals for the inverter switches to control the induction motor. The motor torque is controlled by the quadrature-axis current component and the motor flux is controlled by direct-axis current component. The motor speed is regulated by a PID controller which produces the required torque current component signal.

Table 2. Induction motor parameters

S/No.	Parameter	Symbol	Value
1	Supply Frequency	f	60 Hz
2	Voltage	V	420 V
3	Stator Resistance	Rs	0.288Ω
4	Stator Inductance	Ls	0.0425 H
5	Rotor Resistance	Rr	0.158Ω
6	Rotor Inductance	Lr	0.0418 H
7	Magnetizing Inductance	Lm	0.0412 H
8	Inertia	J	0.4 Kg.m2
9	No. of pole	P	2

2.8 Development of the Simulink Model Diagram

First of all, let us consider the development of a Simulink model for the speed performance or response of an induction motor without controller, the Simulink model for a proportional integral derivative (PID), and lastly, a developed Simulink model for a proportional integral derivative (PID) software tuning mechanism. The speed performance of the induction motor is checked first without any controller and then checked with the help of a proportional integral derivative (PID) controller tuned with software tool. The Simulink model is developed in the MATLAB which is shown in the following Fig.(10), and (11) respectively.

https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

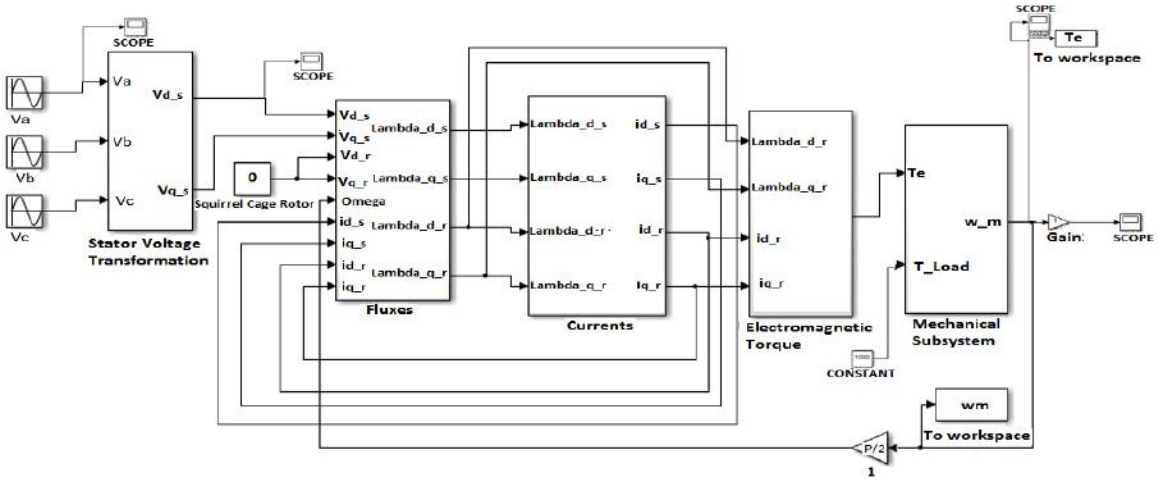

Fig. 10: Developed Simulink model to check speed response of Induction motor without any controller

Using the software tuning mechanism, we can obtain the parameters kp, ki, and kd for optimal performance under the no-load and full-load conditions. In the open loop condition, we can determine the open loop transfer function of the plant by MATLAB program as

$$\mathbf{Gs} = \frac{\mathbf{-4.208e\text{-}12\ s + 0.2598}}{\mathbf{s^2 + 45.99\ s + 0.2981}} \qquad \mathbf{\ldots (46)}$$

Also, the closed-loop system with a PID controller has the following characteristic equation

$$\mathbf{Mc} = \frac{\mathbf{-1.01e\text{-}10\ s^3 + 6.235\ s^2 + 15.85\ s + 0.2598}}{\mathbf{s^3 + 52.23\ s^2 + 16.15\ s + 0.2598}} \qquad \mathbf{\ldots (47)}$$

Equation 46 is obtained by running the MatLab script below and the PID parameters was tuned to give optimal step response as show in fig.(47). The response with controller has a steady state of 17 seconds while the open loop response steady state is 263 seconds.

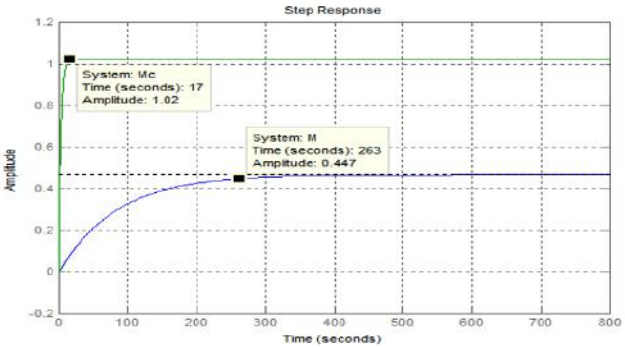

Fig.(11) Open loop and close loop step response of motor transfer function.

The actual motor model "IM_Plant" is controlled using PID controller as shown in fig.(12) below. The parameters of the motor and PID controller are programmed into the system using the MatLab script. This script is used to tune the motor response in the same way as demonstrated in fig.(11) above. Below is the simulation of the motor with controller and the IM_Initiallaization script.

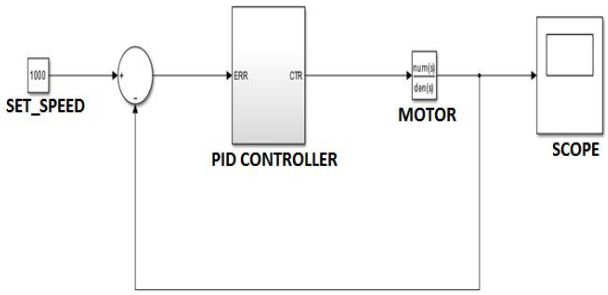

Fig. 12: Developed Simulink model to check speed response of Induction motor with PID controller

III. RESULT

Simulations were carried out in MATLAB environment and the results were verified for the set speed. The speed response of Induction motor is checked for without controller and with PID controller which is shown in Fig.(13) and (14)

https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

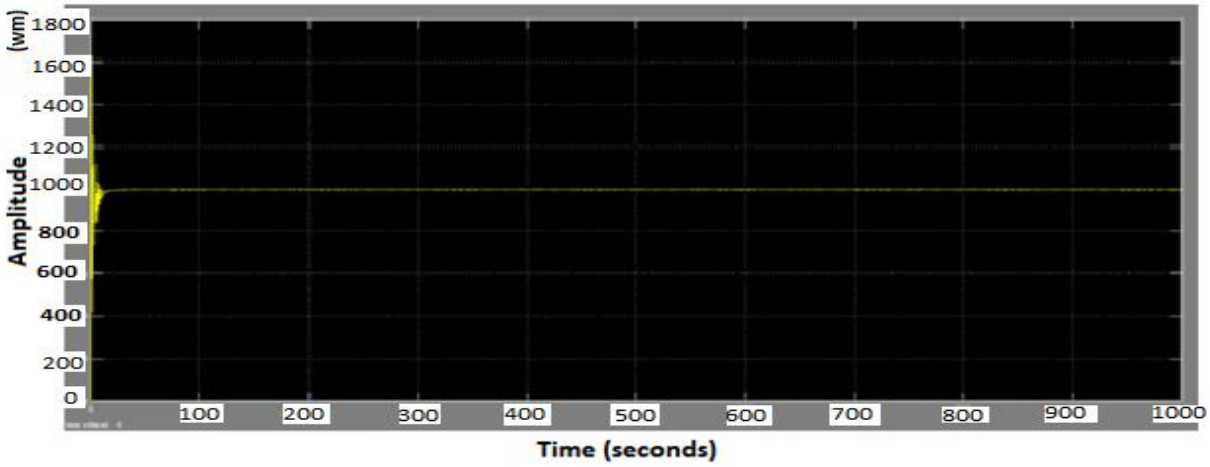

Fig. 13: Speed performance of Induction motor without any controller

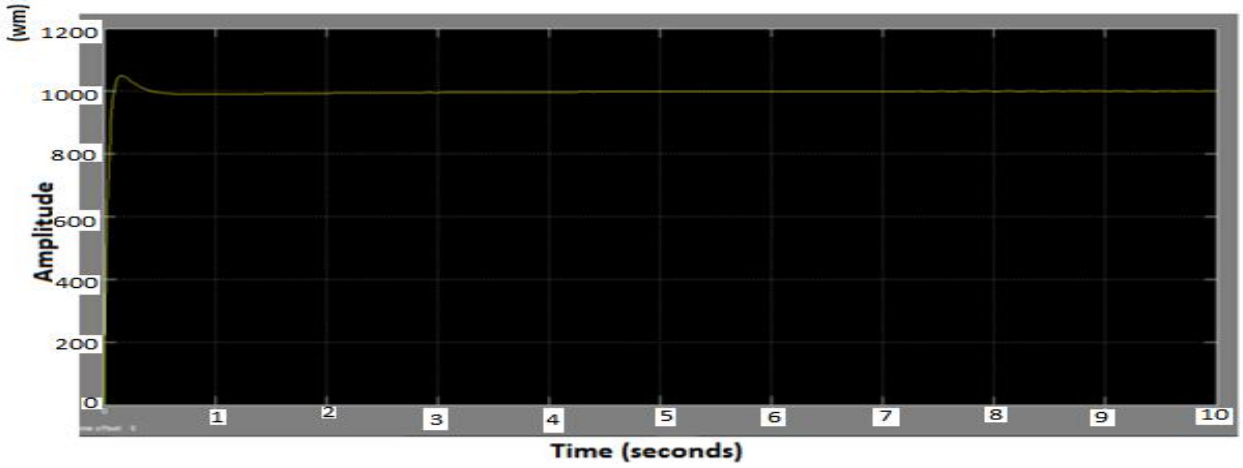

Fig. 14: Speed performance of Induction motor with PID controller

The speed performance of induction motor without any controller is shown in fig.(13). This is for full load condition. When we applied a full load the speed suddenly decreases and is not stable. So as to improve the speed performance, a PID controller tuned with software mechanism. Because of which the steady state error is eliminated and the rise time is improved in figure 4.6. Therefore the PID controller is used to improve the speed performance of Induction motor and the results is shown in fig.(14).

IV. DISCUSSION

The main purpose of this thesis is to control the speed of induction motor using software tuning mechanism of PID controller. From this we come to know that software tuning mechanism of PID controller is a good mechanism for controlling the speed of three phase induction motor. The speed of induction motor using PID controller settled early when using software tuning than other tuning mechanism. As against Zigler – Nicole, the software tools is suitable for any complex system and it is not a trial by error method. It also add the advantage of remote management and flexibility.

V. CONCLUSION

The speed control of an induction motor by software tuning mechanism of PID controllers has led to the conservation of energy and the usage of high performance application such as hybrid vehicles, robotics, wind generation systems, paper and textile mills, and variable-torque centrifugal fan, pump and compressor load applications. Hence there is need to develop a software mechanism for effective speed control for induction motors.

https://dx.doi.org/10.22161/ijaers.4.11.21 *ISSN: 2349-6495(P) | 2456-1908(O)*

REFERENCES

[1] Singh, B. and G. Choudhuri, *Fuzzy Logic Based Speed Controllers For Vector Controlled Induction Motor Drive* IETE Journal of Research 2002. **48**: p. 6.

[2] Z. Q. Zhu, Y.P., D. Howe, S. Iwasaki, R. Deodhar, and A. Pride, *Analysis of Electromagnetic Performance of Flux-Switching Permanent-Magnet Machines by Nonlinear Adaptive Lumped Parameter Magnetic Circuit Model.* IEEE TRANSACTIONS ON MAGNETICS, 2005. **41**(11): p. 11.

[3] Chan, C.C. *The State of the Art of Electric,Hybrid, and Fuel Cell Vehicles*. in *Proceedings of the IEEE* 2007.

[4] ZL, G., *A particle swarm optimization approach for optimum design of PID controller in AVR system.* IEEE Trans Energy Convers, 2004. **19**: p. 91-384.

[5] Chih-Cheng Kao, C.-W.C., Rong-Fong Fung, *The self-tuning PID control in a slider–crank mechanism system by applying particle swarm optimization approach.* Mechatonics, 2006. **16**(8): p. 513-522.

[6] Gaing, Z.-L., *A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System.* IEEE Transactions on Energy Conversion, 2004. **19**(2).

[7] Ang, K.H., G. Chong, and Y. Li, *PID control system analysis, design, and technology* IEEE Transactions on Control Systems Technology, 2005. **13**(4): p. 559-576.

[8] Rujisak Muangsong, D.K., Anak Khantachawana, Panadda Niranatlumpong, *A particle swarm optimization approach for optimal design of PID controller for position control using Shape Memory Alloys* in *Electrical Engineering / Electronics, Computer, Telecommunication and Information Technology*2008, IEEE: Krabi Thailand.

[9] Saghafinia, A., H. Ping, and M. Uddin, *Designing Self-Tuning Mechanism On Hybrid Fuzzy Controller For High Performance And Robust Induction Motor Drive.* International Journal of Advanced Technology & Engineering Research, 2013. **3**: p. 65-72.

[10] R.Arulmozhiyal and K. Baskaran, *Space Vector pulse Width Modulation Based Speed Control of Induction Motor using Fuzzy PI Controller.* international Journal of Computer and Electrical Engineering 2009. **1**: p. 5-7.

[11] Chih-Cheng Kao, C.-W.C., Rong-Fong Fung, *The self-tuning PID control in a slider–crank mechanism system by applying particle swarm optimization approach*, in *Mechanical and Automation Engineering*2006, National Kaohsiung First University of Science and Technology,: Taiwan. p. 513-522.

[12] Liu Y, Z.J., Wang S. *Optimization design based on PSO algorithm for PID controller*. in *Proc Fifth World Congr on Intelligent Control and Automation,*. 2004. Instabul Turkey.

[13] Shi, Y.H. and R.C. Eberthart, *A Modified Particle Swarm Optimizer.* IEEE International Conference on Evolutionary Computation, 1998: p. 7.

[14] Noordin, M.A.B.M., *scaler control of three phase induction motor* 2007, University Teknikal: Malaysia.

[15] Raji, C.T., S.P. Stivastava, and P. Agarwal, *Particle Swarm and Fuzzy Logic Based Optimal Energy control of Induction Motor for a Hoist Load Diagram.* International Journal of Computer Science, 2009. **36**: p. 17-25.

[16] Pillay, P. and R. Krishnan, *Modeling, Simulation, and Analysis of Permanent-Magnet Motor Drives , Part I: The Permanent-Magnet Synchronous Motor Drive.* IEEE Transaction on Industry Application, 1989. **25**(2): p. 9.